5. Users Accounts

Multiple accounts were created on the target system. What are the users?

Example: `Aphelios,Debian-exim,Fido....`

```
┌──(imen㉿hbtn-lab)-[…/webapplicationsecurity/0x0cwebapplication_foresics]
└─$ ./5-users.sh
Aphelios,Debian-exim,Fido,Jax,Nidalee,Senna,dhg,messagebus,mysql,packet,sshd
```

Command Breakdown:

```
grep -E "new user" auth.log | awk '{print $8}' | sed 's/name=//' | sed
's/,$//' | sort | uniq | paste -sd ","
```

1. **`grep -E "new user" auth.log`**:
   - This command searches the `auth.log` file for lines containing the **string "new user"**.
   - The **`-E`** flag enables extended regular expressions, though in this case it's not strictly necessary since you're just searching for the exact string `"new user"`.
   - The goal is to filter log entries that mention the creation of new user accounts.
2. **`awk '{print $8}'`**:
   - After filtering with `grep`, this command uses `awk` to extract the **8th field** from each log entry.
   - In typical `auth.log` entries related to new users, the 8th field usually contains information like `name=<username>`, where `<username>` is the name of the newly created user.
   - `awk '{print $8}'` extracts this field, which contains the user creation information.
3. **`sed 's/name=//'`**:
   - This `sed` command removes the **"name="** prefix from the string.
   - After extracting the 8th field (which might look like `name=<username>`), this `sed` command strips the `name=` part, leaving only the **username**.
4. **`sed 's/,$//'`**:
   - This `sed` command removes the **trailing comma** (if it exists) from the end of the string.
   - It ensures that if there's a comma at the end of the field (which might occur in some logs), it's removed to clean up the extracted usernames.
5. **`sort`**:

    - This command sorts the usernames in **ascending order**.
    - Sorting helps organize the usernames so that duplicates can be easily removed in the next step.
6. `uniq`:
    - After sorting, this command filters out **duplicate entries**.
    - It ensures that only unique usernames are left, removing any repeated occurrences of the same username.
7. `paste -sd ","`:
    - Finally, the `paste` command takes the unique, sorted usernames and concatenates them into a **single line**, with each username separated by a **comma** (`-s` option makes `paste` join the input lines, and `-d ","` specifies that the delimiter between usernames should be a comma).
    - The result is a comma-separated list of unique usernames.

What the command does:

This entire command sequence generates a **comma-separated list of unique usernames** that were associated with "new user" entries in the `auth.log` file. This can be useful for identifying all new users that were created on a system based on the log data.

Example scenario:

Suppose your `auth.log` contains entries like:

```
Feb 24 14:23:56 server sshd[2345]: new user name=user1
Feb 24 14:23:59 server sshd[2345]: new user name=user2
Feb 24 14:24:05 server sshd[2345]: new user name=user1
Feb 24 14:24:10 server sshd[2345]: new user name=user3
```

After running the command:

1. `grep -E "new user" auth.log` will filter the entries that mention "new user", resulting in:

    ```
    new user name=user1
    new user name=user2
    new user name=user1
    new user name=user3
    ```

2. `awk '{print $8}'` will extract the 8th field, which contains:

    ```
    name=user1
    name=user2
    name=user1
    name=user3
    ```

3. `sed 's/name=//'` will remove the `name=` part, leaving:

```
user1
user2
user1
user3
```

4. `sed 's/,$//'` will remove any trailing commas (though in this case, there are none).
5. `sort` will sort the usernames:

```
user1
user1
user2
user3
```

6. `uniq` will remove the duplicate `user1`, leaving:

```
user1
user2
user3
```

7. `paste -sd ","` will concatenate them into a single line:

```
user1,user2,user3
```

Use case:

This command is useful for auditing new user accounts that were created on a system, especially for security or administrative purposes. By collecting all the unique usernames in a single, comma-separated list, you can quickly identify all newly added users.